Catch me if you can: Using Self-Camouflaging Images to Strengthen Graphical Passwords

Mihai ORDEAN
Communications Department
Technical University of Cluj-Napoca
6-28 George Baritiu St, Room 364, Cluj-Napoca, Romania
Email: mihai.ordean@com.utcluj.ro

Karen RENAUD
School of Computing Science
University of Glasgow
18 Lilybank Gardens, Glasgow, G12 8RZ, UK
Email: karen.renaud@glasgow.ac.uk

***Abstract*—In the last decade graphical passwords have been proposed as a viable alternative to the problematical password. One of the most popular of these is the recognition-based graphical password, where the user clicks secret images from one or more *challenge sets* of images, in order to authenticate. While these mechanisms have provable memorability advantages, they are easily as vulnerable to automated sniffing attacks, password-capturing and password computation mechanisms, as are passwords themselves. For example, an attacker can use software to automatically scrape the challenge set images, display these on a duplicate site, and then entice the genuine account owner to reveal the authentication secret. Here we propose a mechanism for addressing this particular weakness of recognition-based graphical passwords. We propose a constantly changing image set, implementing a kind of one-time-password (OTP), which will confound automated attacks by continuously changing the imprint of the secret images.**

It is vital to ensure that the displayable quality of the images is not compromised so that the genuine user can still authenticate without difficulty. Fortunately usability testing showed that the enhanced security model had no impact on the user authentication process. All the benefits of graphical passwords, such as ease of use and increased memorability, are preserved whilst resisting automated attacks.

***Index Terms*—graphical passwords, self-refreshing, recognition, automated attack.**

I. Introduction

Authentication is the process of confirming the identity of a party involved in a digital conversation to the required level of confidence required by the value of the asset being accessed. Credentials are established during an enrollment process, where the party identifies itself and the asset protecting party acknowledges and accepts its credentials. Since there will be a need for the system to be able to confirm the validity of the credentials of any requesting party at a later stage, claims can be authorized using one of the three types of *authentication*, or a combination of two or more of them. The most commonly used mechanism is knowledge-based, where the system and the user share a secret, which the user is required to remember and be able to produce when prompted. This secret is mostly an alphanumeric password. The other two, which are usually used when higher security is required, are token-based (a card or other hardware token), or a biometric. In most cases, therefore, when a user visits the system at a later date, he or she is required to prove that knowledge of the shared secret. If this is done, the user is validated. For most systems passwords are the authentication mechanism of choice. From the software developer's perspective, passwords are an attractive choice: very easy to add to the system, and have well established mechanisms for replacement, and widely used and accepted by end-users. Unfortunately, studies show that most people find alphanumeric passwords difficult to remember [1]. In order for them to cope with and increasing number of passwords they resort to unsafe strategies like choosing simple passwords, reusing passwords in multiple locations, writing passwords down or simply pressing a keyboard sequence in some geometrical order [2]. These "strategies" reduce the security offered by the potentially unlimited size of the alphanumeric password space.

Alternative methods of authentication are proposed to try to address the memorability aspects of passwords, but a good balance between usability, security, and implementation cost (represented by special hardware) has yet to be found. Graphical passwords are a viable candidate through their increased memorability and the fact that they don't require expensive extra hardware or software [3], [4], [5], [6]. Most graphical passwords assign a set of images to the user as their secret, instead of an alphanumeric password. At authentication the user is presented with a grid of images (a challenge set) and he or she confirms knowledge of the "secret" by clicking on the correct images.

Security evaluation for the graphical authentication schemes has proven challenging because most of the testing is done in research environments as opposed to wide-spread Internet usage. Currently there are few platforms that offer commercial grade visual authentication platforms (Passfaces [6]), with the majority of schemes being focused on the research aspects.

A classification done by Biddle *et al.* [7] focuses on four measures relevant towards security of graphical authentication systems:

- theoretical password space
- degree to which user choice issues might weaken security
- the number of probes required to the legitimate server in order to prepare a phishing attack
- the number of user studies that have been conducted with respect to login time and success rate

Camomages, our proposed method, addresses issues affect-

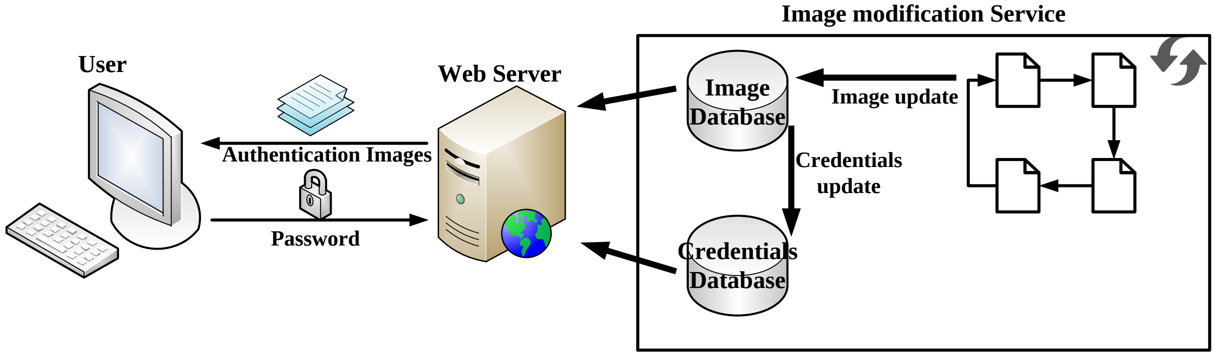

Figure 1: Overview of the application

ing the theoretical password space of the graphical systems. In their research Biddle *et al.* divided the theoretical password space of graphical passwords intro three levels and provided their security equivalent when using an alphanumeric password or tokens. The three levels are: under 20bit (PIN-level equivalent), 20 to 60 bits -password-level equivalent) and over 60 bits (crypto-level, token equivalent) [7]. Recognition based graphical schemes fall in the first category, with security levels similar to PINs. We approached this issue, of increasing this password space, by bolstering the elements from which the password is made, *the password alphabet*, rather than the length of the password itself.

Camomages is a recognition based scheme and as in other recognition based schemes [6], [5] users are required to recognize images rather than text as they would with numbers in a PIN authentication system. However, the difference from other recognition schemes is that we implemented a method to invisibly change data inside images to create a constantly image set. This process does not affect user authentication in any way, as shown by the usability testing described in section IV, but provides unique password alphabets for different logins. We see this as a one-time-password (OTP) authentication [8] in relation to the uniqueness of the password alphabet rather than the uniqueness of the password. This uniqueness of the authentication images makes passwords valid only in the context of a valid alphabet. In short Camomages tackles the problem of the limited password space by requiring a would be attacker to obtain a (password alphabet; password) pair rather than just a password, greatly reducing the threat of automated attacks.

The rest of the paper is organized as follows. The next section presents a functionality overview of Camomages followed by scheme's contributions and threats mitigated. Section III describes our implementation, and the platform used to gather usability data, the actual data being analyzed in following section. We conclude with the limitations of the approach and a summary of the contributions.

II. Camomages, Self-Camouflaging Images

A. Overview

Camomages (figure 1) is a recognition based graphical authentication scheme that provides resistance to automated attacks by performing invisible modifications to images. The purpose is to create temporary image sets indistinguishable from one another. Using these image sets passwords passwords became valid for short periods of time and only with a valid image set. The result is an ever-changing image database that still supports presenting recognizable images to the users for authentication despite changes. The modifications are designed not to affect the visual properties of the files and as such the security layer is invisible to the legitimate users while confounding the attempts of attacking software.

A resident process, independent of the web interface, has direct access to both the user's credentials database and the image database. The purpose of this process is to change the images at specified intervals and maintain the correlation between the new images and enrolled passwords. The passwords are directly derived from the images, as they are constructed from the hashes of the files. Any modification to the image database results in different hashes and an invalid password. This resident process maintains this link to ensure the validity of the passwords. To facilitate password refreshing, the user chosen images are kept in plain text (i.e. the concatenation of their hashes) in the credentials database.

Because security during the transmission is also required, the following password format was chosen:

$$\begin{aligned} PWD = \quad & Hash(Hash(img1)||Hash(img2)||\ldots \\ & \ldots||Hash(imgN)||salt) \end{aligned} \tag{1}$$

where:

|| represents concatenation,
Hash(x) is the output of a hash function applied on the value *x*,
PWD is the transmitted password,
and **salt** is a connection request randomly generated unique value.

We use a salt value that is a randomly generated by the server every connection request. The salt is inserted into the hash of the password to increase its complexity. The resulting hash (i.e. PWD) is only valid for a single connection, preventing replay attacks by sniffing this string.

During authentication all computations are performed at the server side. The received password can be easily recomputed from the server's local values: salt is server generated and stored and the hashes for the images used in user passwords are stored server side.

The following section will analyze the potential attacks on recognition based schemes how these treats are mitigated by Camomages.

B. Attacks mitigated

The combination between the limited password space and probing attacks has always been a problem for the visual authentication schemes [7], mostly because of the technology limitations. Some of the graphic schemes might be more exposed than others (DAS only requires a drawing grid; PassPoints requires one probe; Passfaces requires several probes), but all visual passwords send their *password alphabet* to the user.

Three aspects of password security play a crucial role when analyzing the potential attacks of recognition based schemes: observability, record-ability and guess-ability [4]. The observability directly relates to shoulder-surfing attacks and represents the difficulty with which of the attacker can view the password as it is being entered. Guess-ability represents the difficulty the attacker faces when trying to guess the password. Record-ability represents the ability to record, or capture a password for further use in replay attacks.

Camomages main focus is to enhance resistance to guess-ability and record-ability. The observability aspect of the password security has not been a focus, but some measures have been taken to provide basic security against casual shoulder-surfing. The possible attacks and measures taken are detailed as follows.

Exhaustive search attacks prevention has been one of the main objectives we addressed. The attack implies successively trying all combinations of images until obtaining a valid password is obtained. The limited password length of graphical passwords represents a problem because in comparison alphanumeric passwords. For example, the authentication procedure used by Passfaces [6] consists of asking the user to go through several rounds of selecting a single image belonging to his password from among decoys. The original Passfaces had 4 challenge sets, each composed of 9 images. The guess-ability of the correct combination is one in 9^4 theoretical passwords which is $6561 \approx 2^{13}$ passwords. This password space is smaller than that of a 4 digit PIN.

Our approach to solving this problem was to focus on the authentication image set instead of the length of the password. Camomages implements a method that randomizes the hash values of the images which are then used for building the user's password. This approach allows passwords to be valid only when used with their corresponding image set.

To further clarify, the probability to guess the password is $\frac{1}{\binom{n}{p}}$, where n is the size of the authentication set and p is the length of the password. If on average a brute force attack needs to try half of the total possible passwords the time for a successful password discovery using brute-force is $a_w \approx \frac{1}{2} \cdot \frac{\binom{n}{p}}{d_r}$ where d_r represents the detection rate of passwords over unit of time.

Using the image refreshing can directly control the window of opportunity for the attack, a_w, and keep is at the desired level by modifying the image set. If image refreshing is greater than a_w than we can successfully thwart brute force attacks.

Phishing and replay attacks. Phishing attacks trick users into revealing their credentials by cloning legitimate websites. Usually used in conjunction with phishing attacks, replay attacks use previously harvested credentials to impersonate users.

Camomages create an obstacle for these types of attacks. The temporary validity of the authentication set, allows captured passwords to be valid only for the active image set. Reusing previously captured passwords after the set has been changed would result in an invalid authentication. Camomages authentication scheme also employs the use of a connection-unique server generated value that prevents the use of the same hash value for multiple authentications. One possible workaround for the attacker would be to try and capture the password unencrypted, but even than, direct human intervention would be required to correlate the images between the spoofed ones and the legitimate ones.

Shoulder-surfing attack implies the attacker mounting an observation attack on the authentication process. While not the main focus of our approach, our platform provides sufficient security against casual shoulder surfing [9].

The first approach taken to secure the authentication process against shoulder-surfing was to use a single authentication round. Because images are in a constant state of flux we were able to use a smaller authentication set of 64 images that can be displayed all at once, without severely affecting usability (see section IV). This approach also prevents *intersection attacks* as the attacker is unable to distinguish password contained images and decoys through elimination.

Second, we implemented a randomized position display for the images and no indicator about the selected images. The authentication pattern cannot be detected only by monitoring users actions. For a successful attack recording of both the user actions and the layout of the images is required.

To summarize, Camomages provides increased security against exhaustive search attacks, phishing and replay attacks as well as casual resistance to shoulder surfing attacks. While there are methods to greatly improve the success of the attacks, like image analysis and detection or human outsourcing, we believe that our scheme provides a great improvement to the security of recognition based schemes.

III. Implementation

The chosen graphical authentication scheme was a recognition based scheme. Image refreshing should work with the other graphical schemes, the reason why a recognition scheme was chosen is that this scheme usually has the lowest password space and would benefit the most out of the proposed algorithm. Also usability tests [10] performed show increased memorability and faster login rates (around 20 seconds for passwords consisting of 5 panels of 9 faces) when compared to the other graphical authentication schemes.

Graphical passwords have two options at authentication: either offer users a succession of challenge sets each containing one of the user's secret images, or offer one large challenge set, and require multiple image choices from it. In order to achieve the required security level an entropy greater than 2^{20} is ideally needed. Taking into account interface limitations (too many images depicted simultaneously are hard to be recognized by the user) it was decided to implement an alphabet of 64 images out of which 5 would represent the shared secret, producing an entropy of $\binom{64}{5} = 7624512 \approx 2^{23}$.

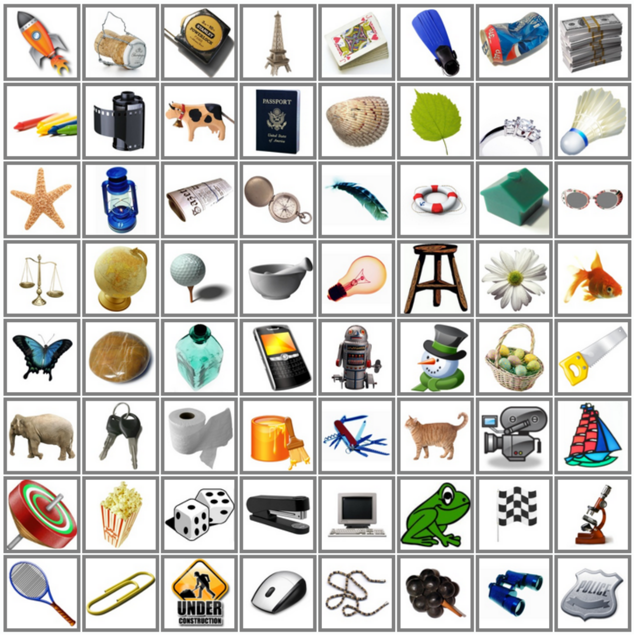

Figure 2: Image set used for authentication

1) Authentication image set: To encourage unpredictable secret image choices, and to support memorability, the image set was constructed using only images depicting simple objects portrayed on white backgrounds. Special consideration was given to avoid obvious patterns such as similar form or object family by reducing the number of elements in and particular semantic group to fewer than 5 (the length of the password).

The testing environment was built from two separate applications: the authentication portal and a OS specific service. Both of this applications share access to a database that stores users credentials and passwords. The application was designed to be fully implemented at the server level, using server based scripting (ColdFusion). The authentication process is performed in the manner described below:

1) The server sends the authentication images to the client each with their attached hash value. In addition to the authentication images the server also sends a randomly generated value to act as unique salt variable for the password.
2) The client concatenates the hashes associated to the images that make up it's password. The salt value is also added to the concatenation and the value is hashed using SHA-2. The hash result is then sent to the server as the authentication password.
3) The server independently computes a hash value created from the password string stored in its database and the previously generated salt variable.
4) The authentication process is successful if the hash values match.

The image hashing is performed using SHA-2 algorithm and it is entirely performed by the server. Database access is shared with the alteration process. This process runs independently from the web interface and generates the hash values from the same image files used in authentication. In the event of the data being intercepted, assigning different hash values to the images by the attacker would still result in an invalid authentication because the server generates its values directly from the files.

A. Image alteration process

The image alteration process is designed to work as a periodic background process installed on the server hosting the authentication interface. The period between runs should be set up with concern to both the security necessities, picture database size and length of the password in order to provide a timespan in which the password cannot be computed even if an attacker gets hold of the picture files. The image alteration is being performed to a raw uncompressed raster map of the image, being independent of the image storage format. The modification process if performed by modifying one of the least significant four bits [11]. Data inserted into the files is randomized to prevent the identification of the modified bits. The modification process is performed as follows:

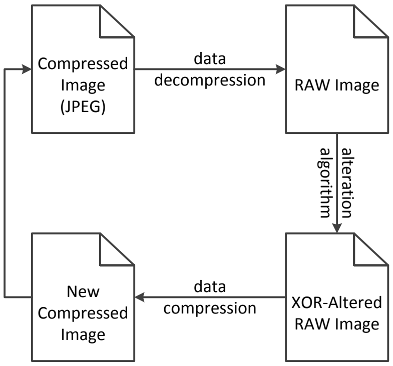

Figure 3: Image alteration process

1) Image files are decompressed into a raw rasterized format.
2) The bits changed are chosen by two random processes, one affecting the chosen bytes to be modified and the other affecting the which bit out of the last 4 LSB.
3) The randomly chosen bits are changed by XOR-ing their values with 1.
4) The resulting raw image is compressed to the original format resulting in an indistinguishable image file with respect to size and perceived image.

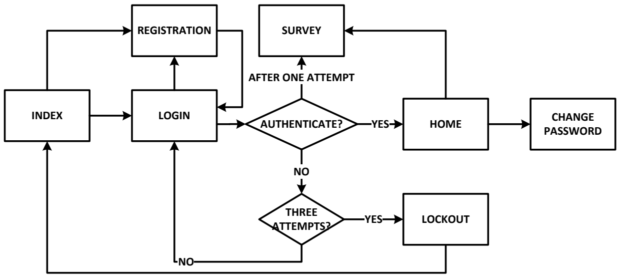

Figure 4: Evaluation Structure

Because of how JPEG compression works, in some cases, some of the altered data might become lost or compensated for and the resulting hash values for the altered images would remain the same as the original. To prevent this scenario and preserve the ability to apply the algorithm over all the files in the database and expect the same results (new hash values) a number of 1024 modifications was needed in every file to render reliable new hash values when using SHA-2 as the hashing algorithm. Tests performed (i.e. graphs detailed in appendix VI-B) showed that randomly choosing 1024 bytes and modifying one random location out of the 4 LSB data provided unique values even after 100k iterations. Given the usual size for images used in this type of authentication (several kilobytes in compressed format) the 1KB required as a minimum file size for the raw uncompressed image can be viewed as small enough to provide a good entropy for alterations.

IV. User Evaluation

A. *Authentication Platform*

The authentication website used for evaluating the usability of the proposed scheme was deployed using server based scripting. The layout is structured into several basic sections: index, login, registration, home, survey and change password. The code for securely displaying the password input is shared among the sections that perform password operations. This interface was implemented using jQuery and creates an embedded pop up within the current section displaying the authentication images in a random order. The password is comprised from five pictures that can be chosen in any order (a sorting is performed prior to sending the password to the server). Each image displayed has an associated hash value tag that represents the actual value that is send back to the server after being combined with the hashes from the other images and a salt value. The salt value itself is transmitted to the client through the HTML code, but it is uniquely generated per page request by the server so it can be seen as measure of protection against connection sniffing and adds to the complexity of the resulting hash value.

During authentication the server independently computes its own hash value using the password and salt values stored in its own databases, and if the value matches the one received authentication is successful. Prior to entering the password, user validation is performed against the database in order to control the login process and prevent uncontrolled image database leakage. After at least one attempt, in order to facilitate data collection, users are granted access to the survey page regardless of their authentication status.

The user study was focused on the evaluation of the usability aspects of the proposed recognition based scheme. The canonical implementation of the authentication procedure is done using a series of panels, each containing one image from the user's password and some different images that act as decoys. Users have to go through several rounds of this type in order to successfully authenticate. The strength of the authentication process is provided by the number of rounds the user has to go through, but this also increases the login time. This process also requires large image databases in order to cope with the required password and decoy images and to provide adequate security against phishing attacks. With the implementation of the security improvements provided by the image alteration process it was possible to reduce the image set required for the authentication to a number of 64 images. Because of the relatively small image set we were able to create a single panel for authentication containing all the images. Security concerns regarding image replication are no longer valid in the context because the images can be viewed as temporary.

In our implementation we were able to obtain a password space of 23 bits by requiring users to select a password made up of 5 order-independent images. The normal password space for a recognition based authentication scheme is between 12 and 16 bits, our scheme providing a password-level security (between 20 and 60 bits) [7] as opposed to the PIN-level security expected from this authentication scheme.

B. *Participants*

The study was conducted in an uncontrolled environment, the participants had access to the web server hosting the authentication portal from their own computers. Relevant data from the login process was gathered independent from the participants input, but there was no control over any of the other activities performed by the participants. A total of 54 participants enrolled in the study out of which 32 completed all the 3 stages, 8 completed the first 2 stages and 14 completed only the enrollment process. The majority of the participants were graduate students from a technical university. The rest of the participants where individuals from the private sector with moderate experience with computers. Out of the total participants 59.25% were male (out of which 51,5% successfully completed all stages) and 40.75% female (out of which 68.1% successfully completed all stages). Age distribution was concentrated in the 25-35 interval with 85.2% participants belonging to this group, 5.55% declaring their age between 35 and 45 and 9.25% as being over 45 years old. Out of the total participants 74.07% declared that they never used graphical authentication before and the rest provided no answer.

C. Procedure

The study was divided into 3 stages: an enrollment process, and two authentication sessions each followed by a short survey. The time between the enrollment process and the first authentication session was at least 3 days, and the login time between the first and the second authentication session was at least 7 days.

The first stage in the data collection was to ask users to provide their age range, gender and enroll with a graphical password. At this moment they had the ability to login to the website and test the functionality of the implemented "change password" interface, but access to the survey was denied. Users were asked not to make notes of their passwords.

Following a 3 day wait period all enrolled users were reminded to login and take the first survey. Regardless of their success rate in the authentication procedure access to the survey was granted and in the event of successful completion a new wait period of 7 days instantiated.

After the 7 days following their successful completion of the first survey enrolled users that finished their first survey were reminded to login for a second time and answer a more extensive survey regarding their experience with graphical passwords. The complete questionnaire is provided in Appendix 1.

D. Results

The purpose of the study was to first evaluate the usability of the scheme and secondly users perception of the security of such scheme if deployed in a real environment. To that end data collected showed very high efficiency in the login process. Three days after the initial enrollment stage 40 users returned to complete the first set of questions and 32 returned after another 7 days (10 days from enrollment) to finish the evaluation.

	after 3 days	**after 10 days**
No. of participants	40	32
Login rates	97,5%	100%
Successful login on the first try	90%	84,37%
Largest no. of tries	3	2
Remembered all img. after 3 tries	97,5%	100%
Smallest no. of img. remembered on 1st try	3	3

Table I
LOGIN RESULTS

Login rates registered were very high with 97,5% users managing to login after 3 days and 100% after 10 days. The statistics described in *Table 1* show promising results both regarding login rates and usability of the scheme discussed.

Image preference between the participants shows that over half of them chose images that were connected with their living environment directly, and the rest was split almost evenly between images visual properties or random images not related in any way. At this point no conclusions can be drawn upon the predictability of the password based on user choices for the used image set.

V. FUTURE WORK

Camomages core concept was to improve the security of a recognition based authentication scheme without affecting the user's experience. To this end the chosen algorithms for modifications only affected the hashes generated from the image files, without affecting visible properties of the images. The data collected by analysis of the obtained hash values for multiple sets of images showed no connection between the image contents, image size and the resulting hash values. However, graphical analysis of images by extracting perceptual hashes was not analyzed and is considered as a vulnerability of our proposed scheme.

Further improvements will focus on developing a method of protecting image identification using computer software to force human intervention during the exhaustive search process which will greatly increase its length.

VI. CONCLUSIONS

The proposed Camomages scheme exploits the fact that virtual objects can indeed carry hidden data. With this in mind it was demonstrated that the security for a recognition based authentication scheme can be greatly enhanced by introducing elements of randomization without affecting the users' authentication experience. Worth mentioning is the fact that our method can be applied to any authentication scheme that uses complex data for authentication.

By analyzing security needs of the deployment environment and its hardware requirements different strategies can be employed when setting the trigger for the self-refreshing algorithm and the lockout policies. The password generated by the use of this scheme can be viewed as a one time password thus making sniffing attacks very hard to perform and, if successful, the sniffed knowledge will be usable only for a very brief period of time. The compromise of the password database storage or the authentication images potentially affects security for a limited window of time. With carefully chosen timings for the image refreshing algorithm trigger this type of attack can be thwarted. Even in the event of the authentication portal being duplicated the generated passwords would be useless when used for authentication in the real environment and the inability to log in would become apparent to the user instantaneously. This scheme provides a number of advantages over the classic approach used when deploying graphical passwords by creating a blend between one time passwords and an easily recognizable PIN type password.

ACKNOWLEDGMENT

This paper was supported by the project "Doctoral studies in engineering sciences for developing the knowledge based society-SIDOC contract no. POSDRU/88/1.5/S/60078, project co-funded from European Social Fund through Sectoral Operational Program Human Resources 2007-2013.

Research was supported by University of Glasgow.

Appendix

A. User Questionnaire

Registration:

Email Address:
Email address of the user (acts as unique identifier).

Name:

Gender:
male/female

Age:
18-25
25-35
35-45
over 45

Stage one:

Have you used Graphical passwords before?
Yes/No

How did the login go?
everything went fine
i had to guess my password
images were very similar
forgot some of the images
forgot my entire password

How many pictures did you remember straight away?
Values from 1 to 5 (all)

Did you have any difficulties?
images were too small
images where too big
there were too many images
images did not load properly
images took a long time to load
I had trouble selecting the images

Stage Two:

How did the login process go this time?
everything went fine
i had to guess my password
images were very similar
forgot some of the images
forgot my entire password

How many pictures did you remember straight away?
Values from 1 to 5 (all)

Did you write down your password to make it easier to remember?
Yes/No

Would you use this method of authentication daily?
values from 0 (highly unlikely) to 9 (highly likely)

How about a couple times a month?
values from 0 (highly unlikely) to 9 (highly likely)

How about if you were to use a graphical password a few times a year?
values from 0 (highly unlikely) to 9 (highly likely)

How did you choose your pictures? Did you choose a common theme, a common colour, or some other way?
I chose random images
I used a common theme
I used a common color
I used a common shape
I used a common class of objects
I used a common pattern (please describe)
I used some other way (please describe)

Comments

B. Hash redundancy

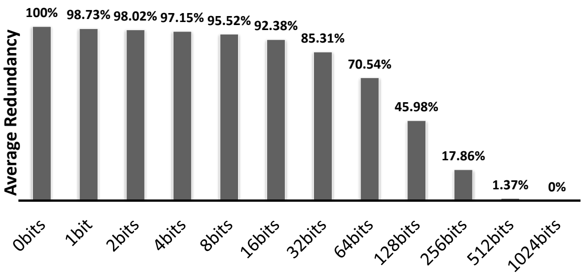

Figure 5: Hash redundancy - JPEG (10k sample size)

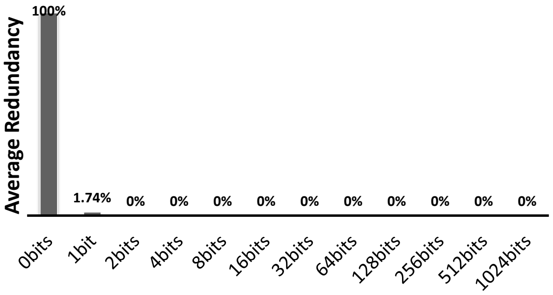

Figure 6: Hash redundancy - BMP (10k sample size)

References

[1] D. Florencio and C. Herley, "A large-scale study of web password habits," in *Proceedings of the 16th international conference on World Wide Web*, pp. 657–666, ACM, 2007.

[2] P. C. V. Oorschot and J. Thorpe, "On predictive models and user-drawn graphical passwords," *ACM Transactions on Information and System Security*, vol. 10, pp. 1–33, Jan. 2008.

[3] S. Brostoff and M. Sasse, "Are Passfaces more usable than passwords: A field trial investigation," in *People and Computers XIV-Usability or Else: Proceedings of HCI*, pp. 405–424, 2000.

[4] A. De Angeli, L. Coventry, G. Johnson, and K. Renaud, "Is a picture really worth a thousand words? Exploring the feasibility of graphical authentication systems," *International Journal of Human-Computer Studies*, vol. 63, pp. 128–152, July 2005.

[5] R. Dhamija and A. Perrig, "Deja vu: A user study using images for authentication," in *Proceedings of the 9th conference on USENIX Security Symposium-Volume 9*, no. 102590, pp. 4–4, USENIX Association, 2000.

[6] Passfaces, "The Science Behind Passfaces The Science Behind Passfaces," *Technology*, pp. 1–5, 2004.

[7] R. Biddle, S. Chiasson, and P. van Oorschot, "Graphical Passwords: Learning from the First Twelve Years," *ACM Computing Surveys (to appear). School of Computer Science, Carleton University*, 2010.

[8] K. Paterson and D. Stebila, "One-time-password-authenticated key exchange," *Information Security and Privacy*, pp. 1–15, 2010.

[9] A. Forget and S. Chiasson, "Shoulder-surfing resistance with eye-gaze entry in cued-recall graphical passwords," *Proceedings of the 28th*, pp. 1–4, 2010.

[10] P. Dunphy and A. Fitch, "Gaze-contingent passwords at the ATM," *4th Conference on*, no. 2, pp. 2–5, 2008.

[11] V. Reddy and A. Subramanyam, "Implementation of LSB Steganography and its Evaluation for Various File Formats," *International Journal*, vol. 872, pp. 173–178, Apr. 2006.